UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/458,883	12/10/1999	THOMAS R. PUZAK	YO999-589	9397

7590 12/17/2002

JAY P. SBROLLINI
IBM CORPORATION
INTELLECTUAL PROPERTY LAW DEPT
PO BOX 218
YORKTOWN, NY 10598

EXAMINER	
WOOD, WILLIAM H	

ART UNIT	PAPER NUMBER
2124	

DATE MAILED: 12/17/2002

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/458,883	PUZAK ET AL.	
	Examiner	Art Unit	
	William H. Wood	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 October 2002* .

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-22* is/are pending in the application.

4a) Of the above claim(s) *2 and 13* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-12 and 14-22* is/are rejected.

7)☒ Claim(s) *21 and 22* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) *4* .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

Claims 1, 3-12 and 14-22 have been examined.

Information Disclosure Statement

1. The Information Disclosure Statement filed on 01 October 2002 has been considered.

Claim Objections

2. Claim 21 is objected to under 37 CFR 1.75 as being a substantial duplicate of claim 10. When two claims in an application are duplicates or else are so close in content that they both cover the same thing, despite a slight difference in wording, it is proper to object to the other as being a substantial duplicate of the allowed claim. See MPEP § 706.03(k).

3. Claim 22 is objected to under 37 CFR 1.75 as being a substantial duplicate of claim 11. When two claims in an application are duplicates or else are so close in content that they both cover the same thing, despite a slight difference in wording, it is proper to object to the other as being a substantial duplicate of the allowed claim. See MPEP § 706.03(k). The claim essentially repeats the limitation of claim 11 from which it depends.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Yeh et al. (USPN 5,742,804).

In regard to claim 1, Yeh taught the following limitations:

i) *a system including a high speed buffer logically placed between memory and at least one processor unit, a method of executing an instruction stream stored in the memory, wherein the instruction stream comprises a sequence of instructions including at least one prefetch instruction that prefetches information from the memory into the high speed buffer* (column 2, lines 27-34)

ii) *deriving first path data from a compiler by analyzing control flow information during compilation, wherein the first path data represents a first path from the prefetch instruction to an instruction that uses information prefetched by the prefetch instruction* (column 4, lines 8-23; column 6, lines 21-27; inserting this instruction indicates the first data had to be derived for the trace parameter of the instruction; the compiler indicates control flow information was analyzed)

iii) *obtaining a branch history defining a path from information generated by branches encountered prior to a subsequent encounter of the prefetch instruction* (column 6, lines 28-37; specifically mentioned is dynamic branch prediction which indicates obtaining branch history of branches happening long before the prefetch is currently encountered)

iv) *generating second path data, wherein the second path data represents a predicted second path of execution* (column 6, lines 28-36; the execution path mentioned here is the second generated path data)

v) *determining whether the first path is consistent with the predicted second path* (column 6, lines 33-36)

vii) *prefetching instructions when the first path is consistent with the predicted second path* (column 6, lines 33-36)

Yeh did not explicitly state prefetching data. However, Yeh did allude to this concept of the prefetch instruction prefetching data to be operated on in column 1, lines 22-25. Here, Yeh states both instructions and data are needed to keep the computer running efficiently. Clearly if the instructions are being prefetched, the data for those instructions would need to be prefetched as well in order to keep up with the continual stream of operation. It would not make sense to have instructions prepared for execution in a cache and then have those same instructions wait for data. If the compiler knows what instructions are going to be executed it can also make a guess as to what data those instructions will need. Therefore, it would have been obvious to one of ordinary skill in the art to implement Yeh with an ability to add a branch predict (data prefetch) instruction in the instruction stream for the purpose of preparing data for use in a predicted future path in order to provide smooth and efficient prefetched data and instruction operations to the cache and thus to the processor.

In regard to claim 2, Yeh taught the additional limitation *wherein the first path data is derived from a compiler performing static compilation* (column 4, lines 49-53; column 6, lines 21-27).

In regard to claim 3, Yeh taught the limitation *wherein the second path data is derived from information characterizing dynamic execution of the sequence of instructions by the processor unit* (column 6, lines 28-33; second path is the execution path).

In regard to claim 4, Yeh taught the limitation *wherein prefetch instruction is added to the instruction stream for execution by the at least one processor unit upon determining that the first path falls within the predicted second path* (column 6, lines 28-37; the unneeded prefetch requests are cancelled and in this manner only the prefetch instruction which is needed is added to the instruction stream for actual execution).

In regard to claim 5, Yeh taught the additional limitation *upon determining that the first path does not fall within the predicted second path, omitting the prefetch instruction from the instruction stream executed by the processor unit* (column 6, lines 28-37).

In regard to claim 6, Yeh taught the limitation *wherein the second path data are associated with one or more branch instructions* (column 6, lines 13-18; column

6, lines 28-33; Figure 1). Yeh does not explicitly state the limitation *wherein the second path data comprises a mask that represents a predicted path of execution that follows the associated branch instructions*. Though, in view of column 6, lines 7-20; column 6, lines 28-37, a masking pattern is being used for the first path, and one of ordinary skill in the art would recognize implementing Yeh's execution path (the second path) in the same bit pattern (i.e. "mask" indicating the path) would make the mentioned comparison operation easier. It would be a simple one-to-one comparison. Therefore, it would have been obvious to implement Yeh with a mask representing the path.

In regard to claim 7, Yeh taught the limitation *wherein the first path data comprises a mask that represents a path of execution from the prefetch instruction to the instruction that uses the information prefetched by the prefetch instruction* (column 4, lines 43-57; column 6, lines 7-20; the bit encodings represent the mask, which represents the path).

In regard to claim 8, Yeh taught the limitation *wherein the second path data are based upon accumulation of predictions associated with branch instructions* (column 6, lines 13-18; column 6, lines 28-33).

In regard to claim 9, Yeh taught the limitation *wherein the second path data is derived from predictions based upon previous execution of the instruction stream* (column 6, lines 13-18; column 6, lines 28-33).

In regard to claims 10 and 21, Yeh taught the additional limitation *wherein the prefetch instruction includes a field that identifies an instruction to prefetch from memory into the high speed buffer* (Figure 1, predict branch 6 instruction shown to have a "target" field; column 4, line 6).

In regard to claims 11 and 22, Yeh taught the additional limitation *wherein the prefetch instruction includes a field that identifies data to prefetch from memory into the high speed buffer* (column 4, line 6; target field identifies the data to be prefetched). Yeh did not explicitly state the limitation *wherein the data is operated on by at least one instruction in the instruction stream.* However, Yeh did allude to this concept of the prefetch instruction prefetching data to be operated on in column 1, lines 22-25. Here, Yeh states both instructions and data are needed to keep the computer running efficiently. Clearly if the instructions are being prefetched, the data for those instructions would need to be prefetched as well in order to keep up with the continual stream of operation. It would not make sense to have instructions prepared for execution in a cache and then have those same instructions wait for data. If the compiler knows what instructions are going to be executed it can also make a guess as to what data those instructions will need. Therefore, it would have been obvious to one of ordinary skill in the art to implement Yeh with an ability to add a branch predict (data prefetch) instruction in the instruction stream for the purpose of preparing data for use in a predicted future path in order to provide smooth and efficient

prefetched data and instruction operations to the cache and thus to the processor.

In regard to claim 12, Yeh taught the following limitations

i) *a system including a memory storing an instruction stream comprising a sequence of instructions including at least one prefetch instruction, a processor unit for executing the sequence of instructions, and a high speed buffer logically placed between the memory and the one processor unit, an apparatus for conditionally executing the prefetch instruction* (column 2, lines 27-34)

ii) *decode logic for deriving first path data from a compiler performing static compilation, wherein the first path data represents a first path from the prefetch instruction to an instruction that uses information prefetched by the prefetch instruction* (column 4, lines 8-23; column 6, lines 21-27; inserting this instruction indicates the first data had to be derived for the trace parameter of the instruction; the compiler indicates control flow information was analyzed)

iii) *logic for obtaining a branch history defining a path from information generated by branches encountered prior to a subsequent encounter of the prefetch instruction* (column 6, lines 28-37; specifically mentioned is dynamic branch prediction which indicates obtaining branch history of branches happening long before the prefetch is currently encountered)

iv) *path prediction logic for generating second path data, wherein the second path data represents a predicted second path of execution* (column 6, lines 28-36; the execution path mentioned here is the second generated path data)

v) *compare logic for determining whether the first path is consistent with the predicted second path* (column 6, lines 33-36)

vi) *execution logic for conditionally prefetching instructions when the first path is consistent with the predicted second path* (column 6, lines 33-36)

Yeh did not explicitly state prefetch logic for prefetching data. However, Yeh did allude to this concept of the prefetch instruction prefetching data to be operated on in column 1, lines 22-25. Here, Yeh states both instructions and data are needed to keep the computer running efficiently. Clearly if the instructions are being prefetched, the data for those instructions would need to be prefetched as well in order to keep up with the continual stream of operation. It would not make sense to have instructions prepared for execution in a cache and then have those same instructions wait for data. If the compiler knows what instructions are going to be executed it can also make a guess as to what data those instructions will need. Therefore, it would have been obvious to one of ordinary skill in the art to implement Yeh with an ability to add a branch predict (data prefetch) instruction in the instruction stream for the purpose of preparing data for use in a predicted future path in order to provide smooth and efficient prefetched data and instruction operations to the cache and thus to the processor.

In regard to claim 13, Yeh taught the limitation *wherein the first path data is derived from a compiler performing static compilation* (column 4, lines 49-53; column 6, lines 21-27).

In regard to claim 14, Yeh taught the limitation *wherein the second path data is derived from information characterizing dynamic execution of the sequence of instructions by the one processor unit* (column 6, lines 28-33; second path is the execution path).

In regard to claim 15, Yeh taught *wherein the execution logic executes the prefetch instruction upon determining that the first path falls within the predicted second path* (column 6, lines 28-37; the unneeded prefetch requests are cancelled and in this manner only the prefetch instruction which is actually needed is then executed).

In regard to claim 16, Yeh taught *wherein the execution logic omits execution of the prefetch instruction upon determining that the first path does not fall within the predicted second path* (column 6, lines 28-37).

In regard to claim 17, Yeh taught the additional limitation *wherein the second path data are associated with one or more branch instructions* (column 6, lines 13-18; column 6, lines 28-33; Figure 1). Yeh does not explicitly state *wherein the second path data comprises a mask that represents a predicted path of*

execution that follows the associated branch instructions. Though, in view of column 6, lines 7-20; column 6, lines 28-37, a masking pattern is being used for the first path, and one of ordinary skill in the art would recognize implementing Yeh's execution path (the second path) in the same bit pattern (i.e. "mask" indicating the path) would make the mentioned comparison operation easier. It would be a simple one-to-one comparison. Therefore, it would have been obvious to implement Yeh with a mask representing the path.

In regard to claim 18, Yeh taught the limitation *wherein the first path data comprises a mask that represents path of execution from the prefetch instruction to the instruction that uses the information prefetched by the prefetch instruction* (column 4, lines 43-57; column 6, lines 7-20; the bit encodings represent the mask, which represents the path).

In regard to claim 19, Yeh taught the limitation *further comprising branch prediction logic for generating predictions associated with branch instructions, and a branch history queue for accumulating the predictions generated by the branch prediction logic, wherein the second path data generated by the path prediction logic is based upon the predictions accumulated in the branch history queue* (column 6, lines 13-18; column 6, lines 28-33). Of particular importance is the branch history queue, which is not explicitly stated in the above referenced passages, however it would have been obvious to one of ordinary skill that such

a structure exists inherently in Yeh since some method of recording the branch histories is needed for prediction purposes.

In regard to claim 20, Yeh taught the limitation *wherein the predictions are based upon previous execution of the instruction stream* (column 6, lines 13-18; column 6, lines 28-33).

Remarks

6. Examiner acknowledges the Declaration is acceptable as the missing addresses are the same as the Post Office addresses.

7. Examiner notes that the amendments to claims 21 and 22 do over come the previous Office Action's rejection under 35 USC § 112. However, upon review of the overall structure of the claim tree for all the claims, Examiner still believes they (claims 21 and 22) would be better suited to being dependent on claim 12.

8. Examiner has considered Applicant's arguments with regard to the double patenting rejection. Examiner does not agree that changing generating to deriving is necessarily different, however the amended claims of application 09/459,739 now include "dynamically" generating. This limitation does not permit Examiner to maintain the double patenting rejection in the current application. Applicant further argues (page 8, lines 20-23) a terminal disclaimer is submitted with application 09/459,739 and that this overcomes the double patenting rejection of claims 10, 11, 21 and 22. This is incorrect procedure. In order to overcome a rejection in the current application a terminal disclaimer must be

present in the current application. Every application must be able to stand alone upon future review and therefore a terminal disclaimer should be included in both applications. However, in this case the point is moot, since claims 10, 11, 21 and 22 depend on independent claims, which are no longer considered rejected under double patenting. Therefore, the double patenting rejections of claims 1, 6, 10-12, 21 and 22 are removed.

9. Examiner has considered Applicant's arguments with regard to claim 1 and found them to not be persuasive. Applicant argues three points: [a)] Yeh fails to indicate the ability to prefetch instructions *and* data, [b)] Yeh does not indicate *deriving first path data from a compiler ... , wherein the first path data represents a first path from the prefetch instruction to an instruction that uses information prefetched by the prefetch instruction*, and [c)] Yeh fails to indicate *obtaining a branch history defining a path from information generated by branches encountered prior to a subsequent encounter of the prefetch instruction.* First, as indicated by Examiner's 103 rejection of claim 11 in the previous office action, Yeh does indicate the ability to prefetch instructions and data. As the rejection stated, prefetching data would come naturally from prefetching instructions in order to keep the processor from having to wait and thus decrease efficiency. Applicant never addressed whether this 103-type rejection was obvious or not. Further proof to the obviousness is the fact that data prefetch instructions are well known themselves as can be seen in the Dubey (USPN 5,774,685) patent included by applicant's original IDS. Second, generating/deriving the first path, wherein the first path represents path data from the prefetch instruction to an

instruction that uses information prefetched by the prefetch instruction, is met by the cited art Yeh. Yeh uses a branch predict instruction to assist in instruction prefetching (column 4, lines 8-9). This means [a)] path data must be generated or derived to know when best to start prefetching and [b)] the path data is from the predict/prefetch instruction to the location where instructions will be assisted by such information being prefetched. Third, Applicant's argument pertains to an added limitation and therefore Examiner's response is the same as set forth above in the rejection of the claim.

10. Examiner has considered Applicant's arguments with regard to claim 3 and found them to not be persuasive. Applicant does not believe that when Yeh states, "... by first predicting (using dynamic and static branch predictors) an execution path of the programmed sequence ..." that this correlates to Applicant's claimed second path. Applicant's second path is a predicted path determined by dynamic execution or in other words based on previous executions a reasonable determination can be made as to what will happen on the current execution pass. This is the exact same concept of Yeh's predicting based on dynamic branch predictors the upcoming execution path.

11. Examiner has reproduced the rejection of the previous Office Action and restructured it to take into account the amendments presented by applicant. Since there are no other direct arguments presented by Applicant, the above rejections are to be taken as Examiner's position on the claims.

Conclusion

12. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to William H. Wood whose telephone number is (703)305-3305. The examiner can normally be reached 7:30am - 5:00pm Monday thru Thursday and 7:30am - 4:00pm every other Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662. The fax phone numbers for the organization where this application or proceeding is assigned are (703)746-7239 for regular communications and (703)746-7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703)305-3900.

William H. Wood
December 16, 2002

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100